

1        **In the Claims**

2        Claims 1-33 are pending and are listed below:

3  
4        1.        (Original) A method comprising:

5        loading one or more source processing chains to support execution of a  
6        development project; and

7        determining whether each of the one or more processing chains will be  
8        subsequently required during execution of this or another development project  
9        and, if so, caching those filter chains which will be subsequently required.

10  
11        2.        (Original) A method according to claim 1, wherein loading of the  
12        one or more processing chains comprises:

13        identifying which source(s) will be required to support execution of the  
14        next M seconds of the development project;

15        searching one or more cache(s) to determine whether the source processing  
16        chain(s) associated with the source(s) are available within the one or more  
17        cache(s); and

18        retrieving the one or more processing chains from a memory location  
19        denoted by an associated one or more pointers in the cache for integration with the  
20        development project.

21  
22        3.        (Original) A method according to claim 2, further comprising:

23        determining whether processing chain(s) retrieved from the cache(s) satisfy  
24        processing requirements of the development project; and

25

1 modifying one or more objects of one or more of the processing chain(s)  
2 retrieved from the cache(s) that do not satisfy the processing requirements of the  
3 development project.

4  
5 4. (Original) A method according to claim 3, wherein modifying one or  
6 more objects may comprise one or more of adding processing objects to the  
7 processing chain(s), removing one or more processing objects from the processing  
8 chain(s), or changing one or more operating attributes of one or more processing  
9 objects within the processing chain(s).

10  
11 5. (Original) A method according to claim 2, wherein M is at least as  
12 long as necessary to construct a processing chain.

13  
14 6. (Original) A method according to claim 1, wherein determining  
15 whether a processing chain will subsequently be required comprises:

16 determining whether any future calls to a source coupled to the processing  
17 chain exist within this development project; and

18 determining whether any future calls to a source coupled to the processing  
19 chain may be received during execution of future development projects.

20  
21 7. (Original) A method according to claim 6, wherein it is assumed that  
22 each processing chain may well be required to support future execution of this or a  
23 future development project.

1 8. (Original) A method according to claim 1, wherein caching the  
2 processing chain comprises:

3 assigning the processing chain a unique identifier; and  
4 storing the unique identifier along with a pointer to a memory location  
5 occupied by the processing chain in a cache.

6  
7 9. (Original) A method according to claim 8, wherein the unique  
8 identifier is one or more of a source file handle, a source file name, a random  
9 numeric identifier uniquely assigned to the processing chain, a graphical icon, an  
10 alphanumeric character, and the like.

11  
12 10. (Original) A storage medium comprising a plurality of executable  
13 instructions which, when executed, implement a method according to claim 1.

14  
15 11. (Original) A computing system comprising:  
16 a storage medium having stored therein a plurality of executable  
17 instructions; and  
18 an execution unit, coupled to the storage medium, to execute at least a  
19 subset of the plurality of executable instructions to implement a method according  
20 to claim 1.

21  
22 12. (Original) A method comprising:  
23 generating a source chain for use in a development project; and  
24 caching the source chain when it is not currently required in the  
25 development project.

1  
2 13. (Original) A method according to claim 12, wherein the processing  
3 chain is cached only if it will subsequently be required in the development project,  
4 or a future development project.  
5

6 14. (Original) A method according to claim 12, wherein caching the  
7 source chain comprises:

8 generating an identifier which is uniquely assigned to the processing chain;  
9 and

10 storing the identifier along with a pointer to memory occupied by the  
11 processing chain in a cache of processing chain pointers.  
12

13 15. (Original) A method according to claim 14, wherein the identifier is  
14 one or more of a source file handle, a file name, an icon, a randomly generated  
15 number uniquely assigned to the processing chain, an alphanumeric identifier, and  
16 the like.  
17

18 16. (Original) A method according to claim 12, further comprising:  
19 identifying a need for a source processing chain; and  
20 retrieving a suitable processing chain from a cache of such processing  
21 chains.  
22

23 17. (Original) A method according to claim 16, further comprising:  
24 integrating the retrieved processing chain into the development project.  
25

1 18. (Original) A method according to claim 16, further comprising:  
2 modifying one or more attributes of the retrieved processing chain before  
3 integration into the development project.

4  
5 19. (Original) A method according to claim 18, wherein modification to  
6 the retrieved processing chain may include one or more of adding processing  
7 objects to the processing chain, removing processing objects from the processing  
8 chain, altering one or more processing characteristics of one or more processing  
9 objects of the processing chain, and the like.

10  
11 20. (Original) A storage medium comprising a plurality of executable  
12 instructions which, when executed, implement a method according to claim 12.

13  
14 21. (Original) A computing system comprising:  
15 a storage medium having stored therein a plurality of executable  
16 instructions; and  
17 an execution unit, coupled to the storage medium, to execute at least a  
18 plurality of the instructions to implement a method according to claim 12.

19  
20 22. (Original) A system comprising:  
21 a plurality of sources; and  
22 an interface, selectively coupled to the plurality of sources, to generate and  
23 implement a development project of processing chains, wherein the interface loads  
24 a processing chain for each of the plurality of media sources at a point during the  
25

1 execution of the project when the chain is required, and wherein the interface is  
2 configured to unload at least a subset of the chains when they are not required.  
3

4 23. (Original) A system according to claim 22, wherein the interface  
5 only loads those processing chains required during the next M seconds of project  
6 execution, and if a current chain-count does not exceed a threshold, T.  
7

8 24. (Original) A system according to claim 23, wherein M is less than a  
9 time required to load a processing chain.  
10

11 25. (Original) A system according to claim 23, wherein if the currently  
12 loaded chain-count has reached a threshold, T, the interface identifies one or more  
13 currently loaded chains that can be unloaded.  
14

15 26. (Original) A system according to claim 25, wherein the interface  
16 identifies one or more currently loaded chains that will not be used during the next  
17 N seconds to unload.  
18

19 27. (Original) A system according to claim 25, wherein the interface  
20 determines whether the identified one or more chains will be required during  
21 subsequent execution of the project, or in a future project and, if so, caches the  
22 identified chain(s).  
23

24 28. (Original) A system according to claim 27, wherein the interface  
25 assigns a unique identifier to processing chains to be cached, and stores the unique

1 identifier along with a pointer to memory wherein the processing chain resides in a  
2 processing chain cache.

3  
4 29. (Original) A system according to claim 22, wherein the interface  
5 removes the identified chains from the active project and caches the removed  
6 chains.

7  
8 30. (Original) A system according to claim 22, wherein the interface  
9 loads processing chains by first searching a cache of processing chains for a  
10 suitable match.

11  
12 31. (Original) A system according to claim 30, wherein if the interface  
13 identifies a suitable match, the processing chain is retrieved from memory for  
14 integration within the processing project.

15  
16 32. (Original) A system according to claim 31, wherein the interface  
17 modifies one or more attributes of the retrieved processing chain before  
18 integration within the processing project.

1 33. (Original) A system according to claim 32, wherein modifying the  
2 processing chain, the interface performs one or more of adding one or more  
3 processing objects to the processing chain, removing one or more processing  
4 objects from the processing chain, modifying one or more processing  
5 characteristics of one or more processing objects within the processing chain.

---

6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25